DEFENSE INFORMATION INFRASTRUCTURE (DII)

COMMON OPERATING ENVIRONMENT (COE)

DII COE Version Description Document
for
Consolidated DCE Application Development Tools v1.1.0.0
(Solaris 2.5.1)

Version Description Document
2/18/97

Table of Contents

Part 1: VDD for DII COE DCE 1.1 Application Development Tools Version 1.0.0.5 (Solaris 2.5.1)

Part 2: VDD for DII COE DCE APIs Version 1.1.0.0 (Solaris)

Part 1: VDD for DII COE DCE 1.1 Application Development Tools Version 1.0.0.5 (Solaris 2.5.1)

1. System Overview

The DCE 1.1 Application Development Tools consists of COTS compilers, library, and header files that allow developers to define and manage a set of programs intended to run in a DCE environment. The development tools support and establish the following methods:

- defining an interface between the component parts
- installing and registering a server so that clients can use it
- set up clients so they can use servers

The DCE 1.1 Application Development Tools are required to develop software which uses DCE resources. Only those systems which have been identified as developers' workstations require the development tools. DCE developers can take advantage of simplified communication between software modules running on different systems. Security can be built into an application which provides a reliable way of determining if a user should be allowed to perform certain functions. The development tools provides the resources that support portability and interoperability with capabilities that hide differences among the various hardware/software platforms.

This delivery consists of COTS software. The COTS software is a licensed product from Transarc called the DCE Application Development Tools.

Updated Tools for the DCE Application Development Tools

Information regarding updates to the development tools are identified in the Release Notes. The Release Notes are maintained and distributed by Transarc when a license is purchased. Once a license is purchased updated Release Notes can be obtained from Transarc's web site, www.transarc.com.

2. Referenced Documents

The following documents are referenced in this Version Description Document:

- OSF® DCE Application Development Guide - Introduction and Style Guide
- OSF® DCE Application Development Guide - Core Components
- OSF® DCE Application Development Reference Volume 1
- OSF® DCE Application Development Reference Volume 2

3. Version Description

3.1 Inventory of Materials Released

- Magnetic Media: Two 8mm tapes consisting of relative tar of the Consolidated DCE Application Development Tools, Version 1.1.0.0/Solaris 2.5.1
    - Transarc DCE 1.1 Application Development Tools v1.0.0.5
    - Simplified DCE APIs v 1.1.0.0
- Version Description Document Defense Information Infrastructure (DII) Common Operating Environment (COE*) Consolidated Application Development Tools Version 1.1.0.0*, February 18, 1997.
- *Installation Instructions for the Consolidated DCE Application Development Tools* (Solaris) Version 1.1.0.0, February 18, 1997

3.2 Software Changes

Reference Appendix A for a list of Transarc DCE 1.1 Application Development Tools software changes.

4. Installation Instructions

Reference the *Installation Instructions for the Consolidated DCE Application Development Tools for Solaris* for instructions on installing the DCE Application Development Tool environment.

5. Known Problems and Errors

There are currently no existing unresolved problems.

6. Release Notes

Information regarding updates to the development tools are identified in the Release Notes. The Release Notes are maintained and distributed by Transarc when a license is purchased. Once a license is purchased updated Release Notes can be obtained from Transarc's web site, www.transarc.com.

Part 2: VDD for DII COE DCE APIs
Version 1.1.0.0 (Solaris)

1. System Overview

The DCE API's are a suite of header files that allow developers to define and manage a set of programs intended to run in a DCE environment. The API's make the DCECOE application development more manageable, configurable and understandable. The DCE API's consists of meta operations, configuration options, and structure.

Developing a DCE application usually requires many similar steps to support the client and server. The API's reduces these redundant steps into integrated steps by using parameter lists. The advantage is a simplified development environment

Updated Tools for the DCE APIs

The following APIs have been updated for the Version 1.1 Consolidated DCE Application Development Tools delivery.

- All API's have been renamed with a DCECOE prefix instead of COEDCE, for consistency.
- The header file coedce.h has been renamed dcecoepublic.h.
- All client-side routines have been made thread-safe. The COEDCEget_vector routine has been eliminated.
- The DCECOEis_authorized routine has been simplified by hiding implementation details within the API and thereby reducing the user setup prior to making the call.
- The new DCECOEstop_server routine allows a client to stop a running server, if authorized.
- The new DCECOEget_attribute routine allows a client to access extended CDS attributes.
- The new DCECOEget_server routine allows a client or server to retrieve it's DCED configuration record.
- Other minor errors have been corrected, as listed in Section 6.

The DCECOE library consists of the routines listed below and described in separate man pages.

Server-side routines:

- DCECOEinitialize_server() - Initializes a DCE server.
- DCECOEsignal_server() - Signal a server to enter listen loop.

- DCECOEcreate_acl() - Creates an access control list (ACL).
- DCECOEis_authorized() - Makes an authorization decision.
- DCECOEfinalize_server() - Terminate server resources.

Client-side routines:

- DCECOEinitialize_client() - Initializes a DCE client
- DCECOEget_attribute() - Retrieves an extended CDS attribute
- DCECOEget_server() - Access the client or server configuration record
- DCECOElocate_server() - Locates a server
- DCECOEinquire_server() - Gets information about a server
- DCECOEstart_server() - Prepare a handle for communications with a server
- DCECOEstop_server() - Requests a running server to stop
- DCECOEfree_servers() - Frees a server vector
- DCECOEfinalize_client() - Frees allocated resources

2. Referenced Documents

The following documents are referenced in this Version Description Document:

- OSF® DCE Application Development Guide - Introduction and Style Guide
- OSF® DCE Application Development Guide - Core Components
- OSF® DCE Application Development Reference Volume 1
- OSF® DCE Application Development Reference Volume 2
- DII DCE Supplemental Consolidated DCE Application Development Tools Programmer's Guide v1.1.0.0

3. Version Description

3.1 Inventory of materials Received

- Magnetic Media: Two 8mm tapes consisting of a relative tar of the DCE APIs Version 1.1.0.0/Solaris 2.5.1
    - Transarc DCE 1.1 Application Development Tools v1.0.0.5
    - Simplified DCE APIs v 1.1.0.0
- *DII DCE Supplemental Consolidated DCE Application Development Tools Programmer's Guide Version 1.1.0.0/Solaris 2.4/2.5.1*
- *Version Description Document Defense Information Infrastructure (DII) Common Operating Environment (COE) DCE APIs Version 1.1.0.0*
- *Installation Instructions for the Consolidated DCE Application Development Tools for Solaris 2.5.1*

3.2 Software Changes

Reference Appendix A for a list of the DCE API software changes.

4. Installation Instructions

Reference the *Installation Instructions for the Consolidated DCE Application Development Tools for Solaris 2.5.1* for instructions on installing the DCE APIs.

5. Known Problems and Errors

- The optional callback parameter in the DCECOEinitialize_server routine has not been fully tested and is not documented in the man page.

6. Release Notes

6.1 Server-side routines

DCECOEinitialize_server()

1. The server now catches fatal signals in order to call DCECOEfinalize_server.
2. An invalid segment name no longer hangs the server.
3. There are no open issues against this API.

DCECOEsignal_server

1. There are no open issues against this API.

DCECOEcreate_acl()

1. There are no open issues against this API.

DCECOEis_authorized()

1. The API is simplified by hiding implementation structures within the API.
2. There are no open issues against this API.

DCECOEfinalize_server()

1. The server now correctly releases binding vector and endpoint entries, preventing delays in server restart.
2. There are no open issues against this API.

6.2 Client-side routines:

DCECOEinitialize_client()

1. The routine now returns an error if the client initializes twice.

2. The routine is now thread-safe.
3. There are no open issues against this API.

DCECOElocate_server()

1. The routine now returns an error if the service or object index is too high, preventing a potential unpredictable error.
2. The routine is now thread-safe.
3. There are no open issues against this API.

COEDCEgetvector()

1. The COEDCEgetvector routine was eliminitated.

DCECOEinquire_server()

1. Corrected a segmentation fault that resulted when count=1 or the number of returned servers was greater than count.
2. The routine is now thread-safe.
3. There are no open issues against this API.

DCECOEstart_server()

1. No longer prints an error message if C_PING fails to find a running server, since this is the normal method of determining if a server is running.
2. The routine is now thread-safe.
3. The callback parameter has not been tested and is not fully documented.
4. There are no other open issues against this API.

DCECOEfree_servers()

1. The routine is now thread-safe.
2. There are no open issues against this API.

DCECOEget_server()

1. There are no open issues against this API.

DCECOEget_attribute()

1. There are no open issues against this API.

DCECOEstop_server()

1. There are no open issues against this API.

Appendix A
Software Changes

This release is the consolidation of Transarc DCE 1.1 Application Development Toolkit and the Simplified DCE API. There are currently no GSPRs open for this delivery.